2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Odometry-based Online Extrinsic Sensor Calibration

Sebastian Schneider, Thorsten Luettel and Hans-Joachim Wuensche

***Abstract*— In recent years vehicles have been equipped with more and more sensors for environment perception. Among these sensors are cameras, RADAR, single-layer and multi-layer LiDAR. One key challenge for the fusion of these sensors is sensor calibration.**

In this paper we present a novel extrinsic calibration algorithm based on sensor odometry. Given the time-synchronized delta poses of two sensors our technique recursively estimates the relative pose between these sensors. The method is generic in that it can be used to estimate complete 6DOF poses, given the sensors provide a 6DOF odometry, as well as 3DOF poses (planar offset and yaw angle) for sensors providing a 3DOF odometry, like a single-beam LiDAR.

We show that the proposed method is robust against motion degeneracy and present results on both simulated and real world data using an inertial navigation system (INS) and a stereo camera system.

I. Introduction

Autonomous vehicles are usually equipped with a large spectrum of different sensors [1], [2], [3]. For sensor fusion the relative position between these sensors has to be known either directly or through a common coordinate system.

The importance of extrinsic sensor calibration is reflected by the amount of literature in this topic, e.g. stereo camera calibration [4] and camera-LiDAR calibration [5]. Recently, researchers have been searching for online calibration methods such as LiDAR to LiDAR calibration [6] or camera to camera calibration [7]. All this work is specifically designed for a certain pair of sensors, in case of LiDAR even certain sensor types.

The work most similar to that presented in this paper comes from Brookshire and Teller [8] who propose an offline method based on 6DOF delta poses of two sensors and least-squares optimization. The extrinsic calibration algorithm presented in this paper also relies on time-synchronized delta poses, but estimates the extrinsic calibration between both sensors online. The generic formulation not only allows to extrinsically calibrate sensors capable of providing 6DOF delta poses. If the sensors provide only planar movement and orientation, such as a single-beam LiDAR, our method at least estimates the 3DOF (planar offset and 1D orientation) pose between such sensors. Even a combination of a 3DOF and a 6DOF odometry sensor for 3DOF pose estimation is possible as shown in the experimental section.

The idea of this paper can be summarized like this: If two rigid sensors, capable of computing their own odometry, were mounted on a moving vehicle in the exact same location with the exact same orientation, then the odometry observed by both sensors would be identical. Due to physical constraints it is more likely that the two sensors are placed adjacent or even further apart. Thus, each sensor perceives the vehicle's movement differently. The main contribution of this paper is to show how this difference in sensor odometry allows for the estimation of the extrinsic calibration between both sensors using an online recursive estimation process.

All authors are with the Institute for Autonomous Systems Technology (TAS), University of the Bundeswehr Munich, 85577 Neubiberg, Germany, Contact author email: sesc@unibw.de

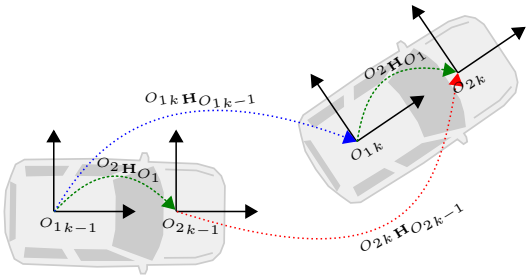


Fig. 1: Coordinate systems O_1 and O_2 of two sensors at times $k-1$ and k. The extrinsic calibration $^{O_2}\mathbf{H}_{O_1}$ (green) is estimated based on the measurements $^{O_1k}\mathbf{H}_{O_1k-1}$ (blue) and $^{O_2k}\mathbf{H}_{O_2k-1}$ (red).

Imagine for example a vehicle equipped with two cameras. One camera points to the front, the other to the left. If the vehicle moves forward the first camera perceives movement along its view direction. However, the second camera perceives the movement as sideways. By using an Unscented Kalman Filter, we can estimate the extrinsic calibration, often also called pose, between both cameras.

This paper is structured as follows: Section II explains how the Unscented Kalman Filter is used to estimate the pose between two odometry-capable sensors. Some applications to simulated and real world data are presented in section III. Finally, section IV concludes the paper and points out future work.

II. Online Extrinsic Calibration

For the estimation of the extrinsic calibration we make use of the Unscented Kalman Filter as it is described in [9]. The equations are summarized as follows:

For a given state vector $\mathbf{x}$, an input vector $\mathbf{u}$ and a parameter vector $\mathbf{p}$ at time k we define the process model as

$$\mathbf{x}_k = f\left(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}\right) \tag{1}$$

and the measurement model as

$$\mathbf{y}_k = g\left(\mathbf{x}_k, \mathbf{p}_k\right). \tag{2}$$

The prediction step of the Unscented Kalman Filter is described through

$$\mathbf{x}_{k-1}^{(i)} = \hat{\mathbf{x}}_{k-1} + (-1)^i \left(\sqrt{(n+\kappa)\mathbf{P}_{k-1}} \right)_{\lfloor \frac{i+1}{2} \rfloor} \tag{3}$$

$$\mathbf{x}^{*(i)}_k = f\left(\mathbf{x}^{(i)}_{k-1}, \mathbf{u}_{k-1} \right) \tag{4}$$

$$\mathbf{x}^*{}_k = \sum_{i=0}^{2n} W_c^{(i)} \mathbf{x}^{*(i)}_k \tag{5}$$

$$\mathbf{P}^*{}_k = \sum_{i=0}^{2n} W_n^{(i)} \left[\mathbf{x}^{*(i)}_k - \mathbf{x}^*{}_k \right] \left[\mathbf{x}^{*(i)}_k - \mathbf{x}^*{}_k \right]^T + \mathbf{Q}_k \tag{6}$$

$$\bar{\mathbf{x}^*}{}_k^{(i)} = \mathbf{x}^*{}_k + (-1)^i \left(\sqrt{(n+\kappa)\mathbf{P}^*{}_k} \right)_{\lfloor \frac{i+1}{2} \rfloor} \tag{7}$$

$$\mathbf{y}^{*(i)}_k = g\left(\bar{\mathbf{x}^*}{}_k^{(i)}, \mathbf{p}_k \right) \tag{8}$$

$$\mathbf{y}^*{}_k = \sum_{i=0}^{2n} W_m^{(i)} \mathbf{y}^{*(i)}_k, \tag{9}$$

where i is an index over the $2n+1$ sigma points $\mathbf{x}^{(i)}_{k-1}$, $\bar{\mathbf{x}^*}{}_k^{(i)}$, and $\mathbf{y}^{*(i)}_k$, that are used to estimate the state vector $\mathbf{x}^*$ and the measurement $\mathbf{y}^*$. The state vector dimension is n. $\mathbf{P}$ denotes the filter's internal covariance matrix, the matrix operator $(\cdot)_i$ extracts the i-th column of a matrix and $\mathbf{0}$ for $i = 0$. The variables $W_c^{(i)}$ and $W_n^{(i)}$ represent the sigma point weights and $\mathbf{Q}$ denotes the process noise covariance matrix. Predicted values are denoted with an asterisk $\cdot^*$, and estimated values with a hat $\hat{\cdot}$.

The measurement update, also called innovation, of the Unscented Kalman Filter is given through the equations

$$\mathbf{P}_{\mathbf{yy}_k} = \sum_{i=0}^{2n} W_c^{(i)} \left[\mathbf{y}^{*(i)}_k - \mathbf{y}^*{}_k \right] \left[\mathbf{y}^{*(i)}_k - \mathbf{y}^*{}_k \right]^T + \mathbf{R}_k \tag{10}$$

$$\mathbf{P}_{\mathbf{xy}_k} = \sum_{i=0}^{2n} W_c^{(i)} \left[\bar{\mathbf{x}^*}{}_k^{(i)} - \mathbf{x}^*{}_k \right] \left[\mathbf{y}^{*(i)}_k - \mathbf{y}^*{}_k \right]^T \tag{11}$$

$$\mathbf{K}_k = \mathbf{P}_{\mathbf{xy}_k} \mathbf{P}_{\mathbf{yy}_k}{}^{-1} \tag{12}$$

$$\hat{\mathbf{x}}_k = \mathbf{x}^*{}_k + \mathbf{K}_k \left(\mathbf{y}_k - \mathbf{y}^*{}_k \right) \tag{13}$$

$$\mathbf{P}_k = \mathbf{P}^*{}_k - \mathbf{K}_k \mathbf{P}_{\mathbf{yy}_k} \mathbf{K}_k^T, \tag{14}$$

with the measurement noise covariance matrix $\mathbf{R}$.

Imagine two rigid sensors mounted at coordinate systems O_1 and O_2 as illustrated in figure 1. We define the homogeneous transformation matrix that transforms from O_1 to O_2 as ${}^{O_2}\mathbf{H}_{O_1}$, which is the extrinsic calibration that we seek to find.

We treat the vehicle movement as a discrete-time process from time $k-1$ to k, such that the odometry measured by each sensor can be expressed through the homogeneous transformation matrices ${}^{O_{1k}}\mathbf{H}_{O_{1k-1}}$ and ${}^{O_{2k}}\mathbf{H}_{O_{2k-1}}$ respectively.

Both sensors are supposed to be rigid. Thus the extrinsic calibration remains constant over time. We provide three solutions to the problem of finding ${}^{O_2}\mathbf{H}_{O_1}$ using the Unscented Kalman Filter.

A. Estimating ${}^{O_2}\mathbf{H}_{O_1}$

In 3D space homogeneous transformation matrices depend on 6 pose parameters: The position $(x, y, z)^T$ and the orientation represented as Euler Angles $(\Phi, \Theta, \Psi)^T$, roll, pitch and yaw. For a given homogeneous transformation matrix ${}^b\mathbf{H}_a$ we define ${}^b\mathbf{P}_a$ to be the vector of the corresponding 6 pose parameters, as well as the functions

$$\mathbf{H}({}^b\mathbf{P}_a) = {}^b\mathbf{H}_a \tag{15}$$

$$\mathbf{P}({}^b\mathbf{H}_a) = {}^b\mathbf{P}_a\,. \tag{16}$$

For this first solution let $\mathbf{x}$ be the state vector containing exactly the 6 pose parameters of ${}^{O_2}\mathbf{P}_{O_1}$

$$\mathbf{x} = (x, y, z, \Phi, \Theta, \Psi)^T\,. \tag{17}$$

Since we are estimating a rigid transformation between both sensors, the state vector's components are supposed to be almost constant over time, only changing with the estimation approaching the desired extrinsic calibration. The process model function of equation 1 reflects this through the following definition

$$f\left(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}\right) = \mathbf{x}_{k-1}. \tag{18}$$

For the measurement model we assume that the noise of sensor 1 can be neglected (we lift this assumption later) and set the parameter vector $\mathbf{p}_k = \mathbf{P}\left({}^{O_{1k}}\mathbf{H}_{O_{1k-1}} \right)$ to the first sensor's measured odometry. The measurement model function of equation 2 becomes

$$g\left(\mathbf{x}^*{}_k, \mathbf{p}_k\right) = \mathbf{P}\left(\mathbf{H}\left(\mathbf{x}^*{}_k\right)^{-1} \cdot \mathbf{H}\left(\mathbf{p}_k\right) \cdot \mathbf{H}\left(\mathbf{x}^*{}_k\right) \right). \tag{19}$$

Less formally speaking, we predict what we would have to measure as the second sensor's odometry ${}^{O_{2k}}\mathbf{H}^*{}_{O_{2k-1}}$ assuming the current extrinsic calibration (the state vector estimate from equation 18) was correct and the first sensor's odometry was free of noise.

Even though ${}^{O_{1k}}\mathbf{H}_{O_{1k-1}}$ is considered noise free (which in reality most likely is not true), we can at least define the measurement noise for ${}^{O_{2k}}\mathbf{H}_{O_{2k-1}}$ through the measurement noise covariance matrix $\mathbf{R}$. The experiments in section III show how this affects the overall filter quality.

B. Estimating ${}^{O_2}\mathbf{H}_{O_1}$ and ${}^{O_{2k}}\mathbf{H}_{O_{2k-1}}$

In order to enhance the previous approach we incorporate one sensor's odometry into the state vector

$$\mathbf{x} = \left(\mathbf{x}_{odom}, \mathbf{x}_{extr}\right)^T \tag{20}$$

$$\mathbf{x}_{odom} = \left(x_{odom}, y_{odom}, z_{odom}, \Phi_{odom}, \Theta_{odom}, \Psi_{odom}\right)^T \tag{21}$$

$$\mathbf{x}_{extr} = \left(x_{extr}, y_{extr}, z_{extr}, \Phi_{extr}, \Theta_{extr}, \Psi_{extr}\right)^T. \tag{22}$$

While the process model for $\mathbf{x}_{extr}$ remains the same as in section II-A

$$\mathbf{x}^*_{extr\,k} = \hat{\mathbf{x}}_{extr_{k-1}}, \tag{23}$$

we compute the part of the state vector $\mathbf{x}_{odom}$ based on the input vector $\mathbf{u}_k = \mathbf{P}\left({}^{O_{1k}}\mathbf{H}_{O_{1k-1}}\right)$, which we set to the measured odometry of sensor 1

$$\mathbf{x}^*_{odom\,k} = \mathbf{P}\left(\mathbf{H}\left(\hat{\mathbf{x}}_{extr_{k-1}}\right)^{-1} \cdot \mathbf{H}\left(\mathbf{u}_k\right) \cdot \mathbf{H}\left(\hat{\mathbf{x}}_{extr_{k-1}}\right)\right). \tag{24}$$

This simplifies the measurement model, since our measurement is now part of the state vector

$$g\left({\mathbf{x}^*}_k, \mathbf{p}_k\right) = \mathbf{x}^*_{odom\,k}. \tag{25}$$

Through this change we have not yet integrated both sensor's noise into the filter. However the noise of sensor 1 now manifests itself in the uncertainty of $\mathbf{x}^*_{odom\,k}$ as defined in the upper left 6×6 matrix of $\mathbf{Q}$. This part reflects the uncertainty of the second sensor's odometry $\mathbf{x}^*_{odom\,k}$ induced by the noise of the first sensor's odometry. Compared the the previous version of the filter this allows at least for some consideration of the first sensor's noise even though its noise cannot be grasped directly.

C. Integrating the Measurement Noise of Both Sensors

In order to integrate the first sensor's noise correctly into the filter of the previous section, we have to find a way to transform the sensor's noise through the process modell into the process noise covariance matrix $\mathbf{Q}$ in equation 6.

However, simply setting the process covariance for $\mathbf{x}^*_{odom}$ to the noise of the first sensor ignores the fact, that both sensors might be placed at different locations and with different orientation. Assume again a pair of sensors pointing to a vehicle's front and left side. A high noise in the first sensors forward direction has to be considered as lateral noise for the second sensor, and not in its forward direction.

Fortunately, the Unscented Kalman Filter already provides a solution to this problem: Equations 3 to 6 describe the so called Unscented Transform. In equation 3 the $2n+1$ sigma points are drawn symmetrically to approximate the normal distribution $\mathcal{N}\left(\hat{\mathbf{x}}_{k-1}, \mathbf{P}_{k-1}\right)$. Keeping the input vector $\mathbf{u}_{k-1}$ fixed, these sigma points are propagated through the process model in equation 4 and combined in equation 5. Then equation 6 integrates the propagated sigma points, the state vector estimate and the process noise covariance matrix into the estimate for the internal filter covariance matrix $\mathbf{P}^*$.

It turns out, that with slight modifications the Unscented Transform allows to propagate the input vector's noise in the same way. In a few words, instead of calculating sigma points for the state vector and propagating these sigma points with fixed $\mathbf{u}_{k-1}$ through the process model, we keep the state vector fixed and calculate the $2m+1$ sigma points for $\mathbf{u}_{k-1}$, where m is the dimension of the input vector, using the input vector's covariance $\mathbf{S}$.

$$\mathbf{u}^{(i)}_{k-1} = \mathbf{u}_{k-1} + (-1)^i \left(\sqrt{(m+\kappa)\mathbf{S}_{k-1}}\right)_{\lfloor \frac{i+1}{2} \rfloor} \tag{26}$$

$${\mathbf{x}_{\mathbf{u}}}^{*(i)}_k = f\left({\mathbf{x}^*}_k, \mathbf{u}^{(i)}_k\right) \tag{27}$$

$${\mathbf{x}_{\mathbf{u}}}^*_k = \sum_{i=0}^{2m} W^{(i)}_c {\mathbf{x}_{\mathbf{u}}}^{*(i)}_k \tag{28}$$

$${\mathbf{Q}_{\mathbf{u}}}^*_k = \sum_{i=0}^{2m} W^{(i)}_n \left[{\mathbf{x}_{\mathbf{u}}}^{*(i)}_k - {\mathbf{x}_{\mathbf{u}}}^*_k\right]\left[{\mathbf{x}_{\mathbf{u}}}^{*(i)}_k - {\mathbf{x}_{\mathbf{u}}}^*_k\right]^T \tag{29}$$

Obviously, the dimension of the propagated sigma points ${\mathbf{x}_{\mathbf{u}}}^{*(i)}_k$ is n, the dimension of the state vector, and not m. Thus, the covariance matrix ${\mathbf{Q}_{\mathbf{u}}}^*_k$, which essentially is the estimated process noise induced by the paramteter vector $\mathbf{u}_k$ with covariance matrix $\mathbf{S}_k$ given the current state vector estimate ${\mathbf{x}^*}_k$, also has dimension $n \times n$.

On a closer look at our process model in equations 23 and 24 we see that the prediction of the extrinsic calibration does not depend on $\mathbf{u}$. Thus, propagating the sigma points $\mathbf{u}^{(i)}_k$ through the process model will yield neither variance nor covariance with that part of the state. Therefore the estimated covariance matrix ${\mathbf{Q}_{\mathbf{u}}}^*_k$ has the following structure

$${\mathbf{Q}_{\mathbf{u}}}^*_k = \begin{pmatrix} {\mathbf{U}^*}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \tag{30}$$

and ${\mathbf{U}^*}_k$ is the $m \times m$ covariance matrix that reflects how the input noise covariance $\mathbf{S}_k$ affects the state vector estimate, more precisely ${\mathbf{x}^*}_k$.

We cannot simply replace $\mathbf{Q}_k$ in equation 6 with ${\mathbf{Q}_{\mathbf{u}}}^*_k$, because this would assume zero noise for the state vector part $\mathbf{x}_{extr}$. Instead we suggest to add ${\mathbf{Q}_{\mathbf{u}}}^*_k$ to the result of equation 6.

III. Experiments

We have conducted several experiments with both simulated and real world data to compare and verify the above algorithms.

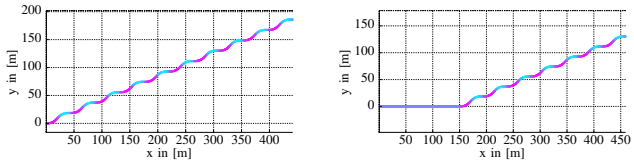
A. Simulations

For easier comparison the ground truth extrinsic calibration $\mathbf{x}_{extr}$, used for generating the simulation data, remains the same

$$\mathbf{x}_{extr} = (1, 1, 1, 0.1, 0.1, 0.1)^T \tag{31}$$

throughout all experiments. This places the second sensor one meter away in each direction and rotates the sensor $5.73°$ around each axis. Two distinct datasets form the basis of the evaluation, all generated with the same virtual vehicle having a wheel base of 3.5 m. In the first dataset the vehicle is driving a slalom course. The second dataset shows a combination of straight driving followed by slalom driving. We refer to the filter described in section II-A as *Filter 1*, the one described in section II-B is called *Filter 2* and the version of section II-C we name *Filter 3*.

1) Slalom Dataset: In this dataset the simulated vehicle drives a slalom course with a sinusoidal steering angle (amplitude $\lambda_{max} = 10°$, frequency $f_\lambda = 0.1\,\text{Hz}$) and resulting roll angle (amplitude $\Phi_{max} = 3°$, frequency $f_\Psi = 0.1\,\text{Hz}$). The vehicle's velocity is constant at $v = 5\,\frac{\text{m}}{\text{s}}$. Simulated measurements are taken every 0.1 s. The resulting track consisting of 30 000 pairs of poses is illustrated in figure 2a.



(a) Slalom dataset (b) Mixed dataset

Fig. 2: Simulated vehicle path of the slalom (a) and mixed (b) datasets (first 1000 poses). The vehicle's roll angle is color coded. High roll angles are represented in magenta, low (negative) roll angles in cyan.

For each simulated measurement we add 0-mean gaussian noise to the positional coordinates with a variance of $\sigma^2_{\text{pos}} = 1 \times 10^{-5}$ and to the angular coordinates with a variance of $\sigma^2_{\text{ang}} = 3 \times 10^{-6}$ according to our real world inertial navigation system (INS). The noisy sequence of all pose pairs is fed into all three filters. The initial state vectors of all filters are initialized with $\mathbf{x}_{\text{init}} = \mathbf{0}$, the covariance matrices are initialized as diagonal matrices with diagonals as described in table III. Figure 3 shows the evolution of each filter's extrinsic calibration component of its state vector.

By comparing the plots for the slalom dataset in figure 3 we see that all three filters are able to eventually find the correct extrinsic calibration. *Filter 1* is the slowest of all three regarding convergence, followed by *Filter 2* and *Filter 3*. The z component of the extrinsic calibration seems to be the hardest to find, because for each of the three plots the z component converges last. This is expected considering the low roll angle amplitude $\Phi_{max} = 3°$ and is reflected in a higher value at the corresponding components of the filter's internal covariance matrix $\mathbf{P}$. Another effect worth mentioning is that the correct values for the angular components are found at least 4 to 6 times faster than the positional components. The angular components are easier to find, since even small angular errors magnify during the vehicle's movement. This is beneficial since for data fusion an accurate orientation of sensors usually takes precedence over an accurate position. Apart from that the three filters differ in how accurate they estimate the individual components of the extrinsic calibration. The standard deviations and the mean squared error (MSE) of the extrinsic calibration over the last 1000 data points for each filter are shown in table I.

	σ_x	σ_y	σ_z	σ_Φ	σ_Θ	σ_Φ
F. 1	0.0002	0.0010	0.0072	0.0147	0.0737	0.0111
F. 2	0.0029	0.0034	0.0176	0.0607	0.2519	0.0677
F. 3	0.0006	0.0005	0.0013	0.0223	0.0155	0.0197

	MSE_x	MSE_y	MSE_z	MSE_Φ	MSE_Θ	MSE_Φ
F. 1	0.0000	0.0007	0.0010	0.0000	0.0002	0.0005
F. 2	0.0003	0.0005	0.0027	0.0001	0.0011	0.0003
F. 3	0.0000	0.0001	0.0001	0.0001	0.0000	0.0000

TABLE I: Standard deviations and MSE of the extrinsic calibration estimation in the slalom dataset. Units are [m] for positional components and [°] for angular components.

The standard deviations of *Filter 2* are considerably higher than those of the other two. *Filter 3* has comparable or lower standard deviations in the magnitude of mm for positional components and 100^{th}-° for angular components. Except for the z component the MSEs are close to zero.

2) Mixed Dataset: The mixed dataset is composed of 30 iterations of a 30% straight course followed by 70% slalom course with the same parameters as for the slalom dataset. Thus, the dataset also consists of 30 000 pairs of poses with the same amount of noise added as before. The resulting track is visualized in figure 2b.

All three filters converge in a similar manner as for the slalom dataset, thus figure 4 only shows the first two iterations through the mixed dataset, i.e. 2000 pairs of poses.

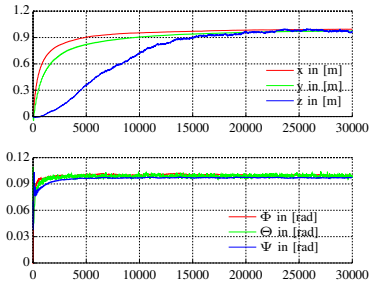
The highlighted regions in figure 4 correspond to the straight driving parts of the dataset (data points 0 to 300 and 1000 to 1300). Brookshire and Teller [8] describe the so called motion degeneracy as trajectories out of which the extrinsic calibration cannot be inferred, i.e. is not observable. Straight driving is an example for such a trajectory. This can be seen e.g. in figure 4a: The datasets starts with straight driving. Since the positional components of the extrinsic calibration are not observable the filter does not improve the estimation on these parameters. In fact the corresponding values of the internal process covariance matrix $\mathbf{P}$ increase, indicating the increasing uncertainty in these estimates. However, the angular components are observable, so the filter tries to find suitable values that satisfy the estimates of the positional components. As soon as the slalom movement starts (data point 301) all components become observable and the estimates of all extrinsic calibration parameters start to converge to the ground truth. At data point 1000 the estimates have not yet reached their true value, but the vehicle again goes into straight driving. While the positional components remain basically unchanged due to their unobservability, the filter still improves on the angular components. This process goes on until the filter finally converges. Similar behaviour can be seen for the two other filters in figures 4b and 4c. They only differ due to the state vector now also containing the odometry $\mathbf{x}_{odom}$ measured by one sensor. The standard deviations and the MSE of the extrinsic calibration over the last 1000 data points for each filter are shown in table II.

	σ_x	σ_y	σ_z	σ_Φ	σ_Θ	σ_Φ
F. 1	0.0007	0.0012	0.0084	0.0098	0.0735	0.0065
F. 2	0.0042	0.0040	0.0225	0.0367	0.2363	0.0479
F. 3	0.0003	0.0006	0.0008	0.0137	0.0171	0.0131

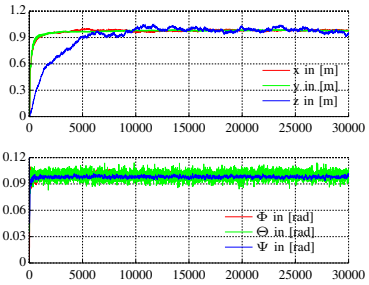
	MSE_x	MSE_y	MSE_z	MSE_Φ	MSE_Θ	MSE_Φ
F. 1	0.0002	0.0016	0.0407	0.0000	0.0002	0.0004
F. 2	0.0004	0.0008	0.0448	0.0000	0.0010	0.0002
F. 3	0.0000	0.0000	0.0006	0.0000	0.0000	0.0000

TABLE II: Standard deviations and MSE of the extrinsic calibration estimation in the mixed dataset. Units are [m] for positional components and [°] for angular components.

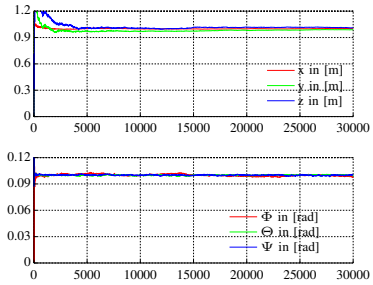
Compared to the slalom dataset we see that the introduction of motion degeneracy increases the uncertainty of the z-component. This coincides with our previous observation that in this setting z is difficult to estimate. Apart from that we see, that the mixed dataset yields estimates with similar uncertainties. Thus, using an Unscented Kalman Filter to



(a) Calibration results of *Filter 1*.

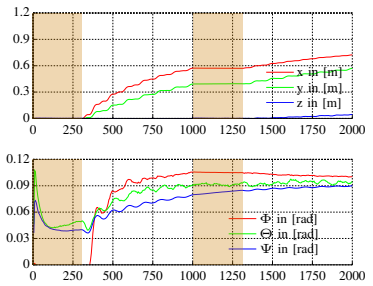


(b) Calibration results of *Filter 2*.

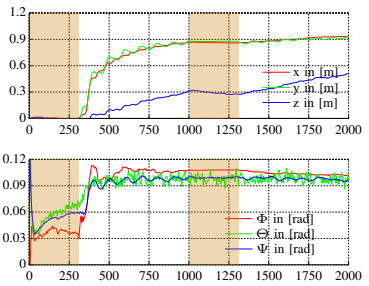


(c) Calibration results of *Filter 3*.

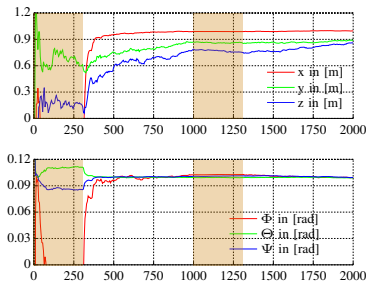
Fig. 3: Evaluation of the slalom dataset.



(a) Calibration results of *Filter 1*.



(b) Calibration results of *Filter 2*.



(c) Calibration results of *Filter 3*.

Fig. 4: Evaluation of the mixed dataset.

Filter 1	$\mathbf{x}_{\text{init}}$	0	0	0	0	0	0				–		
	$\mathbf{P}_{\text{diag}}$	1	1	1	1	1	1				–		
	$1 \times 10^5 \cdot \mathbf{Q}_{\text{diag}}$	1	1	100	0.3	120	0.3				–		
	$1 \times 10^1 \cdot \mathbf{R}_{\text{diag}}$	3	3	3	1	1	1				–		
Filter 2	$\mathbf{x}_{\text{init}}$	0	0	0	0	0	0	0	0	0	0	0	0
	$\mathbf{P}_{\text{diag}}$	1	1	1	1	1	1	1	1	1	1	1	1
	$1 \times 10^4 \cdot \mathbf{Q}_{\text{diag}}$	100	100	100	3	3	3	1	0.1	5	0.03	105	0.3
	$1 \times 10^2 \cdot \mathbf{R}_{\text{diag}}$	1	1	1	0.3	0.3	0.3				–		
Filter 3	$\mathbf{x}_{\text{init}}$	0	0	0	0	0	0	0	0	0	0	0	0
	$\mathbf{P}_{\text{diag}}$	1	1	1	1	1	1	1	1	1	1	1	1
	$1 \times 10^7 \cdot \mathbf{Q}_{\text{diag}}$	0.01	0.01	0.01	0.003	0.003	0.003	10000	100	100	30	30	30
	$1 \times 10^9 \cdot \mathbf{R}_{\text{diag}}$	1	1	1	0.3	0.3	0.3				–		
Filter 3 (real data)	$\mathbf{x}_{\text{init}}$	0	0	0	0	0	0	0.78	-0.12	0.42	-3.142	0	0
	$1 \times 10^2 \cdot \mathbf{P}_{\text{diag}}$	100	16	16	0.09	0.0625	0.0625	25	25	25	12.25	75.69	12.25
	$1 \times 10^6 \cdot \mathbf{Q}_{\text{diag}}$	0.1	0.1	0.1	0.3	3	0.3	0.1	0.1	0.1	3	3	3
	$1 \times 10^4 \cdot \mathbf{R}_{\text{diag}}$	10000	0.01	0.01	0.09	0.0625	0.25				–		

TABLE III: Initial values and variances for the three filters for all experiments.

estimate the extrinsic calibration shows to be robust against motion degeneracy.

3) Combining a 3DOF and a 6DOF odometry sensor: To demonstrate the flexibility of our algorithm we also simulate the calibration of a planar sensor, such as a single-beam LiDAR only perceiving odometry in terms of x, y and Ψ, and a full 6DOF sensor, like an inertial measurement unit. Adjusting *Filter 3* to such a sensor turns out to be rather simple. Since the LiDAR perceives its movement only in x, y and Ψ, we change equation 21 to

$$\mathbf{x}_{odom} = (x_{odom}, y_{odom}, \Psi_{odom})^T . \tag{32}$$

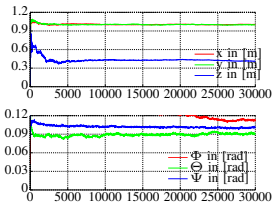
Finally, in equation 24 where the LiDAR odometry is predicted, we can simply ignore the predictions of z_{odom}, Φ_{odom} and Θ_{odom}.

The calibration results in figure 5 show that the filter is capable of estimating the correct values for x_{extr}, y_{extr} and Ψ_{extr} in both datasets. Interestingly, the filter also provides estimates for the other parameters. This is caused by effects like the vehicle's roll inducing a slight lateral movement of the simulated LiDAR. Since these effects are hardly noticable and in some cases even ambigious, the filter is not able to turn them into correct estimates.

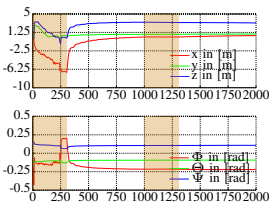
4) Simulating different sensor arrangements: Additionally to the presented simulations we have generated multiple slalom datasets with other extrinsic calibrations $\mathbf{x}_{extr}$. While each filter finds the correct calibrations, *Filter 3* requires less to none adjustments to $\mathbf{Q}_{\text{diag}}$ and $\mathbf{R}_{\text{diag}}$ compared to both other filters.

B. Real Dataset

In the real dataset our vehicle MuCAR-4 is driving a slalom course as illustrated in figure 6 consisting of 1795 pose pairs. The course has a total length of approximately 1500m. The sensors are an OxTS RT3003 INS placed below the arm rest near the vehicle's center of gravity and a



(a) Calibration results on the slalom dataset.



(b) Calibration results on the mixed dataset.

Fig. 5: Evaluation of *Filter 3* for a 3DOF and a 6DOF odometry sensor pair.

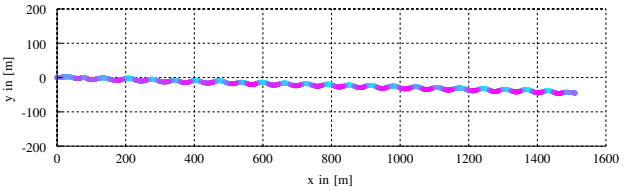


Fig. 6: Vehicle path of the real dataset. The vehicle's roll angle is color coded. High roll angles are represented in magenta, low (negative) roll angles in cyan.

Bumblebee XB3 stereo camera mounted upside down below the rear view mirror. While the INS already provides delta poses, the motion of the camera is obtained using the visual odometry library VISO2 [10].

We initialize the filter with the values of table III taking the roughly known extrinsic calibration into account: E.g. we know that the camera is rotated upside down, so the roll angle is initialized with $-\pi$.

The calibration results are shown in figure 7. We see that the filter almost already converges after the first 1000 data points. We obtain standard deviations for the extrinsic calibration as shown in table IV.

	σ_x	σ_y	σ_z	σ_Φ	σ_Θ	σ_Φ
Filter 3	0.0040	0.0094	0.0161	0.2865	0.2636	0.3552

TABLE IV: Standard deviations of the extrinsic calibration estimation in the real dataset. Units are [m] for positional components and [°] for angular components.

Compared to the simulations the uncertainties for the estimation on real data are higher, but should still be good enough for most applications. However, care has to be taken when defining $\mathbf{x}_{\text{init}}$ and the covariance matrixes, since – depending on your sensor configuration – the filter turns out to be very sensible to these values and might converge into a local minimum, which – at least in our tests – are easily distinguishable from the expected results. As in the simulation results the angular components converge faster than the positional ones.

IV. Conclusion

We have addressed the online extrinsic calibration problem between two sensors using a recursive estimation process. The only requirement for the proposed method is that both

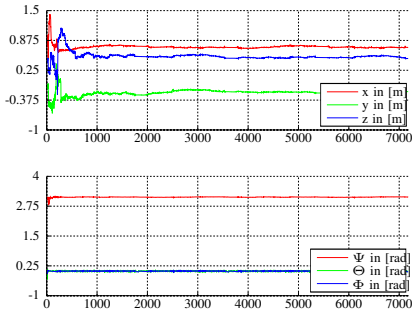


Fig. 7: Calibration results of *Filter 3* on the real dataset.

sensors are able to measure their odometry, e.g. calculate synchronized delta poses. Starting with a straight forward solution using the Unscented Kalman Filter, we improved this solution in two steps, evaluating our models on two simulated datasets. Thereby we have also derived a solution for the integration of parameter noise into the estimation process using the Unscented Transform.

The resulting filter has been applied to real world data obtained from an INS and a stereo camera system. Not only was the filter able to estimate the extrinsic calibration between the INS and the camera system, is has also shown to be robust against motion degeneracy.

Our future work will focus on the question of what kind of motion is required to obtain an accurate calibration and thereby avoiding filter convergence into local minima. This includes an observability analysis of the state vector, incorporating alternative representations for the Euler angles such as quaternions.

References

[1] D. Göhring, M. Wang, M. Schnürmacher, and T. Ganjineh, "Radar/Lidar Sensor Fusion for Car-Following on Highways," in *Proc. IEEE Int'l Conf. on Automation, Robotics and Applications (ICARA)*, Dec. 2011, pp. 407–412.

[2] M. Himmelsbach, F. von Hundelshausen, T. Luettel, *et al.*, "Team MuCAR-3 at C-ELROB 2009," in *Proc. of 1st Workshop on Field Robotics, Civilian European Land Robot Trial*. Oulu, Finland: University of Oulu, 2009.

[3] C. Urmson, J. Anhalt, D. Bagnell, *et al.*, "Autonomous Driving In Urban Environments: Boss and the Urban Challenge," *Journal of Field Robotics*, vol. 25, no. 1, pp. 425–466, June 2008.

[4] Z. Zhang, "Flexible Camera Calibration By Viewing a Plane From Unknown Orientations," in *Proc. IEEE Int'l Conf. on Computer Vision*, vol. 1, Corfu, Greek, Sept. 1999, pp. 666–673.

[5] S. Schneider, M. Himmelsbach, T. Luettel, and H.-J. Wuensche, "Fusing Vision and LIDAR - Synchronization, Correction and Occlusion Reasoning," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, San Diego, CA, USA, June 2010, pp. 388–393.

[6] C. Gao and J. R. Spletzer, "On-line Calibration of Multiple LIDARs on a Mobile Vehicle Platform," in *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, Anchorage, AK, USA, May 2010, pp. 279–284.

[7] T. Dang, C. Hoffmann, and C. Stiller, "Continuous Stereo Self-Calibration by Camera Parameter Tracking," *IEEE Trans. Image Process.*, vol. 18, no. 7, pp. 1536–1550, 2009.

[8] J. Brookshire and S. Teller, "Extrinsic Calibration from Per-Sensor Egomotion," in *Proc. of Int'l Conf. on Robotics: Science and Systems (RSS)*, Sydney, Australia, July 2012.

[9] E. A. Wan and R. van der Merwe, "The Unscented Kalman Filter," in *Kalman Filtering and Neural Networks*, S. Haykin, Ed. New York: Wiley, 2001, pp. 221–280.

[10] A. Geiger, J. Ziegler, and C. Stiller, "StereoScan: Dense 3d Reconstruction in Real-time," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, Baden-Baden, Germany, June 2011, pp. 963–968.